

■ Find Unique Element Using XOR – DSA Notes

■ Problem Statement

Given an array where every element appears twice except for one number, your task is to find the single element that appears only once. Use an efficient approach with XOR.

■ Examples

Example 1:

Input: nums = [2, 3, 5, 4, 5, 3, 4]

Output: 2

Example 2:

Input: nums = [2, 3, 5, 3, 2]

Output: 5

Example 3:

Input: nums = [1, 1, 2, 2, 3]

Output: 3

Example 4:

Input: nums = [7]

Output: 7

■ Core Logic – XOR Property

1. XOR of two identical numbers is 0 $\rightarrow a \wedge a = 0$
2. XOR of any number with 0 is the number itself $\rightarrow a \wedge 0 = a$
3. Therefore, XORing all array elements cancels duplicates, leaving the unique number.

Efficient: $O(n)$ time, $O(1)$ space

■ Dry Run (nums = [2, 3, 5, 3, 2])

Step	Element	ans (XOR Result)
Start	-	0
1	2	2
2	3	$2 \wedge 3 = 1$
3	5	$1 \wedge 5 = 4$
4	3	$4 \wedge 3 = 7$
5	2	$7 \wedge 2 = 5$ (Unique)

■ C++ Code

```
class Solution {
public:
    int findUniqueElement(vector &nums) {
        int ans = 0;
        for (int n : nums) {
            ans ^= n;
        }
        return ans;
    }
};
```

■■ Time & Space Complexity

Time Complexity: $O(n)$ — single traversal

Space Complexity: $O(1)$ — no extra memory used

■ Key Notes

- XOR is the best method for this problem.
- Avoid sorting or maps (slower).
- Perfect for interview one-liners.